Securing FIPA agent communication

Niklas Borselius and Chris J. Mitchell
Information Security Group
Royal Holloway, University of London
Egham, Surrey, TW20 0EX, UK
{Niklas.Borselius, C.Mitchell}@rhul.ac.uk

Abstract— **The agent paradigm has been the subject of much research during the last decade. Recently, security of multi-agent systems has gained increased attention. In this paper we consider the FIPA agent communication specifications from a security perspective, and outline how security functionality can be added.**

Keywords— **security, multi-agent system, communication, FIPA.**

I. Introduction

FIPA[1] is a non-profit organisation promoting the use and development of intelligent agents by openly developing specifications supporting interoperability among agents and agent-based applications. FIPA's specification for agent communication has become a defacto standard. Security has not been a driving force for research and development of Multi-Agent Systems (MASs) and therefore has only received limited attention from the agent research community. Although, an earlier, today obsolete, FIPA specification did consider security for agent communication [1], security is not fully covered in the current specifications [2]. It appears that the FIPA specifications and the general state of agent research are now mature enough to require security services for agent communication. FIPA has recognised the need for improved security and initiated work in the area[2].

In this paper we will evaluate the FIPA specifications from a security point of view and propose extensions to the specifications in order to provide security services for agent communication. We will address the following security services for agent message communication:

- **Integrity**, protects against improper modification of a message;
- **Origin authentication**, the corroboration that the source of data received is as claimed;
- **Confidentiality**, guarantees that unauthorised parties cannot access information in transit.

Non repudiation, which can be considered a *stronger version* of origin authentication, will not be further addressed in this paper. Non repudiation (of data origin and data reception) can be achieved using authentication mechanisms in combination with timestamps and a third party to settle disputes.

For the purpose of this paper we are assuming a multi-agent system in an *open environment*, that is, a system where no single authority is in control of the system, which means that agents (and other entities) cannot be assumed to act in a perfectly honest manner. We are also assuming a supporting Public Key Infrastructure (PKI) for management of certified cryptographic public keys. The precise requirements for a PKI for an open multi-agent system is a research topic on its own.

The paper is structured as follows. In the next two sections we briefly describe the FIPA agent communication specifications, first the communication model and then the message structure. Section IV describes the Open PGP message format and how this can be applied to agent communication. In section V we consider the required interaction between

[1]Foundation for Intelligent Physical Agents, see `http://www.fipa.org`

[2]See `http://www2.elec.qmul.ac.uk/~stefan/fipa-security` for the state and progress of this work.

an agent and its platform if the platform is to provide secure communications services to the agent. Finally, we conclude and point out future work in section VI.

II. The FIPA communication model

Figure 1 shows the FIPA message transport reference model [2].

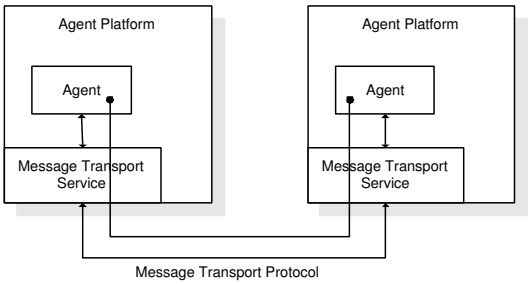

Fig. 1. FIPA message transport model

The message transport service is a service typically provided by the agent platform on which an agent is executing. However the agent and the message transport service do not have to be located on the same host. The message transportation service supports the transportation of messages between agents on any given agent platform and between agents on different platforms through the provision of an Agent Communication Channel (ACC). FIPA recognises three options for an agent when sending a message to another agent residing on a remote platform (illustrated in figure 2).

1. Agent A sends the message to its local ACC using a proprietary or standard interface. The ACC then takes care of the transmission of the message to the correct remote ACC. The remote ACC will then eventually deliver the message.
2. Agent A sends the message directly to the ACC on the remote agent platform on which B resides. This remote ACC then delivers the message to B.
3. Agent A sends the message directly to agent B, using a direct communication mechanism. The message transfer, including buffering of messages and any error messages, must be handled by the sending and receiving agents. (No further specification for this communication mode is covered by FIPA.)

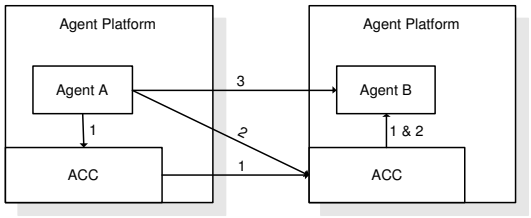

Fig. 2. Methods of communication between agents on different platforms

III. FIPA Message structure

In this section we will describe the structure of a FIPA message [2] and then consider how security is addressed.

A message is made up of a message envelope, containing transport information, and a message body comprising of the agent communication data or ACL (Agent Communication Language) message. Table I shows the structure of the message envelope. An ACC should deliver the whole message, including the message envelope, to the receiving agent. However, it is possible for agent platforms to provide middleware layers to free agents from the task of processing the envelope [2].

The envelope parameter called encrypted is optional, and if used indicates that the message is encrypted as defined in RFC 822 [3]. The syntax for the parameter is two words. The first word indicates the software used to encrypt the body, and the second, optional, word is intended to aid the recipient in selecting the proper decryption key. However, current FIPA platforms based on JADE (Java Agent DEvelopment Framework) [4], FIPA-OS [5], etc., do not support RFC 822.

Table II shows the structure of an ACL message. The performative element is the only mandatory element of an ACL message; all other elements are optional. The ACL message structure does not include any security specific elements.

Parameter	Description
to	Names of primary recipients of the message, mandatory.
from	Name of the agent who sent the message, mandatory.
comments	Optional comment.
acl-representation	Name of the syntax representation of the message body, mandatory.
payload-length	The length of the message body, optional.
payload-encoding	The language encoding of the message body, optional.
date	Message creation date and time, added by the agent, mandatory.
encrypted	Indication how the message body has been encrypted, optional.
intended-receiver	The name of the agent to whom this instance of a message is to be delivered, optional.
received	Time received by an ACC, optional.
transport-behaviour	Transport requirements of the message, optional.

TABLE I
MESSAGE ENVELOPE DESCRIPTION

A. Security evaluation

The only provision for security in the FIPA message structure is the envelope primitive: encrypted. This allows for additional software to be used to offer message confidentiality through encryption. Origin authentication and message integrity are not provided for. Furthermore, the encrypted field in the envelope is intended for the ACC, not the agent itself, which means that the encryption would be under the control of the ACC, not the agent.

IV. Open PGP

In this section we will briefly describe the Open PGP message structure in order to be able to evaluate its appropriateness for FIPA messages.

Open PGP [6] is a non-proprietary protocol for protecting email using public key cryptography. It is based on PGP as originally developed by Phil Zimmermann [7]. The Open PGP protocol defines standard formats for encrypted messages, signatures, and certificates for exchanging public keys. Over the past decade, PGP, and more recently Open PGP, have become well used de facto standards for encrypted email. By becoming an IETF standard [6], Open PGP may be implemented freely.

PGP messages are constructed from a number of records referred to as packets. A packet is a piece of data that has a tag specifying its meaning. A PGP message consists of a number of packets. Some of those packets may themselves contain other packets. Each packet consists of a packet header, followed by the packet body. The packet header has a tag which denotes what type of packets the body holds. Table III shows the defined tags.

As can be seen from Table III, PGP supports the secure message services described in section I, including message confidentiality, through the use of symmetric and asymmetric encryption algorithms, and message integrity and origin authentication, through the use of digital signatures.

Some of the PGP packets listed in the ta-

Tag no.	Description
0	Reserved - a packet tag must not have this value
1	Public-Key Encrypted Session Key Packet
2	Signature Packet
3	Symmetric-Key Encrypted Session Key Packet
4	One-Pass Signature Packet
5	Secret Key Packet
6	Public Key Packet
7	Secret Subkey Packet
8	Compressed Data Packet
9	Symmetrically Encrypted Data Packet
10	Marker Packet
11	Literal Data Packet
12	Trust Packet

TABLE III
OPEN-PGP PACKET TYPES

Element	Description
performative	The type of communicative act of the message.
sender	Name of the agent who sent the message, mandatory.
receiver	Names of primary recipients of the message, mandatory.
reply-to	Indicates that subsequent messages in this conversation are to be directed to the agent named in this element.
content	Denotes the content of the message.
language	Denotes the language in which the content element is expressed.
encoding	Denotes the specific encoding of the content language expression.
ontology	Denotes the ontology used to give a meaning to the symbols in the content expression.
protocol	Denotes the interaction protocol that the sending agent is employing with this message.
conversation-id	Used to identify the ongoing sequence of communicative acts that together form a conversation.
reply-with	Introduces an expression that will be used by the responding agent to identify this message.
in-reply-to	Denotes an expression that references an earlier action to which this message is a reply.
reply-by	Denotes a time which indicates the latest time by which the sending agent would like to have the reply.

TABLE II
ACL MESSAGE ELEMENTS

ble are designed for key management. This is functionality we have not considered, but which is crucial for deployment of cryptography on a large scale.

PGP is not the only standard for describing cryptographically protected message content. Another example is Cryptographic Message Syntax (CMS) (RFC 3369) [8]. CMS is used by S/MIME (Secure/Multipurpose Internet Mail Extensions) [9] and specifies syntax for representing digitally signed, hashed, authenticated, or encrypted arbitrary message content (CMS is based on PKCS #7 [10]). Similar functionality to that described for Open PGP is available in CMS.

A. Using PGP for FIPA messages

We will now consider how the PGP message structures can be used with FIPA messages.

PGP can be used to encrypt and sign the entire ACL message without making any changes to the message envelope. This would leave the encoding and decoding of PGP structured messages for the agent to take care of. This would not require any changes to the FIPA specifications, assuming the agent is able to determine if a message is PGP encoded or not (this might, for example, be achieved by using the encoding field).

If there are reasons for treating the elements within the ACL differently, for example to only encrypt or sign certain elements, additional information would need to be provided to the agent to indicate how the message should be processed.

If the ACC service is to perform encryption/decryption and process signatures 'seamlessly', additional information needs to be provided in the message envelope.

The advantage with using an existing standardised protocol such Open PGP or CMS is that it is already defined, thereby avoiding duplication of work. The drawback might be that the already standardised protocol has features unnecessary for our purpose, possibly adding unnecessary payloads and, perhaps, confusion.

To summarise, it is rather straightforward

to make use of the Open PGP message structure for FIPA agent communication. However, in order to allow for all the communication modes described in section II, as well as for cryptographically protected messages, the specifications should allow encryption/decryption and signature processing to be carried out by the ACC service as well as by the agent. This would require additional information to be added in the message envelope. For full flexibility, the ACL message should also carry information describing the security mechanisms applied to the message. The information exchange between an agent and ACC service is further described in the next section.

V. Agent — platform interaction

In this section we consider further where the communication security mechanisms fit in the FIPA architecture. We will also highlight architectural issues not covered in the previous sections, namely key management and security policies.

Leaving the complete coding and encoding of encrypted or signed messages to the agent may not always be desirable. To keep agents simple and small, it can be more efficient to let the executing host deal with this potentially complex task. This can be done in different ways, as described in the next two paragraphs.

As shown in figure 3, one way is to let the agent receive the message as usual, and when the agent has determined that it is an encrypted or signed message it forwards the message to the appropriate service to decrypt it or to validate the signature. The agent would then be returned the decrypted message or an indication of the outcome of the signature verification. Similarly when the agent wants to send an encrypted message or a signed message it would send the message to the appropriate service and be returned the processed message, which now can be sent as usual.

A second option, depicted in figure 4, would be to let the ACC intercept the communication and offer the security services in a more transparent way to the agent. This would require the ACC to be able to determine if an incoming communication is encrypted or signed, as well as getting an indication from the agent whether encryption or signing should be done before sending the message.

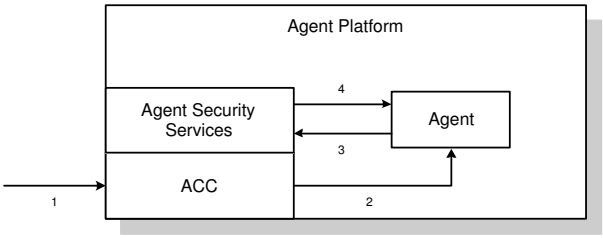

Fig. 3. Security services separated from ACC

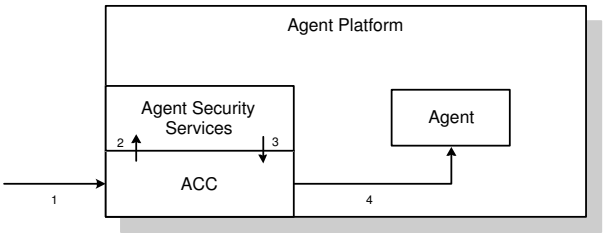

Fig. 4. Secure communication services offered 'transparently' to the agent

Both modes of operation should be provided to ensure that the communication modes described in section II are fully supported.

There are two important, and non-trivial, issues we have not yet covered. One is key management. When encryption is used the encryption/decryption keys need to be managed. For asymmetric cryptography a PKI is typically used to facilitate certain aspects of the key management. In a multi-agent system various key management issues arise due to the fact that the agent executes on a host which in theory has full control over the agent, even more so if the agent depends on the host to carry out processing involving the cryptographic keys. The following questions need to be considered:

- Is the agent in control of its own key, or is this completely/partly delegated to the platform where the agent is executing?
- Does the agent need its own keys, or can agents on the same host use the same crypto-

graphic keys?

Since applications will have different requirements we believe that agents should have the freedom to decide on these issues themselves, rather then only supporting one approach.

The second important thing we have not yet discussed is that of security policies. If an agent depends on the host to perform cryptographic tasks, the agent needs to communicate its requirements to the host. Likewise the host needs to let the agent know certain information about its processing.

While PGP might be sufficient for the carrying of encrypted and signed messages, further specifications need to be developed for a complete solution. Communications between the agent and the executing host need to be standardised to cover the transfer of key management and security policy data. In the remaining of this section we consider the information that need to be exchanged between an agent and the ACC service, which we assume is residing on the platform where the agent is executing.

For outgoing communication (communication originating from the agent) the agent needs to be able to supply (explicitly or implicitly) the platform with the following information:

- request message encryption;
- supply an encryption key or indicate that the platform's encryption key should be used;
- choice of encryption algorithm (and other possible algorithm related options, including key length, padding, etc.);
- request message signing;
- supply a signature key or indicate that the platform's signature key should be used;
- choice of signature algorithm (and possible algorithm related options, including key length, padding, etc.).

If an agent requests encryption or signing, the platform would typically apply some default values for parameters not specified by the agent.

For incoming communication (communication destined for the agent) the platform needs to be able to inform the agent regarding the following:

- If the message was protected through encryption during transit;
- if so, indicate method of encryption that was applied to the message (e.g. encryption algorithm and key length);
- if the message carried a digital signature;
- if so, to what extent the signature has been verified.

The platform might need certain information from the agent in order to decrypt a message or verify a signature including the following:

- decryption key;
- trusted Certification Authorities (CAs) and agents;
- signature verification requirements (e.g. verification against revocation lists and max length of certification chains).

The above information can be exchanged between the agent and platform in various ways. The most straightforward way appears to be to let the agent supply message specific information with every message that is passed to the ACC service. Another option, which may be more efficient, is for the agent to have a complete security policy description that can be passed to the ACC service prior to any other communication taking place. Such a policy should be allowed to be as complex as might be required. Different requirements might, for example, apply depending on the destination of a message. A third option is to combine the other two options. An agent can then carry, and supply to the platform, a complete communication security policy, but can also request to have a particular message treated differently by supplying specific information with the message.

VI. Conclusions and future work

The FIPA agent communication specifications are lacking sufficient functionality to provide secure communication. By using an existing message structure such as the Open PGP message format, sufficient protection can be achieved for the communication. We have considered where security services can

be applied to agent communication within the FIPA architecture, and described the information exchange required between an agent and the ACC security services. Further detailed analysis and specification is however required for a complete solution.

Another way to achieve secure agent communication appears to be the XML (Extensible Markup Language) specifications. A Document Type Definition (DTD) to carry an ACL message has been defined by FIPA. Various efforts are in progress for specifying how cryptographic services can be applied to XML. These are also likely to provide the security services required for agent message communication.

Acknowledgments

The work reported in this paper has formed part of the Software Based Systems area of the Core 2 Research Programme of the Virtual Centre of Excellence in Mobile & Personal Communications, Mobile VCE, `www.mobilevce.com`, whose funding support, including that of the EPSRC, is gratefully acknowledged. More detailed technical reports on this research are available to Industrial Members of Mobile VCE.

References

[1] *FIPA 98 Specification Part 10, Version 1.0, Agent Security Management*, Geneva, Switzerland, October 1998, Obsolete.

[2] *FIPA Agent Message Transport Service Specification, Document no. XC00067D*, Geneva, Switzerland, August 2001.

[3] David H. Crocker, *Standard for the format of ARPA Internet text messages, IETF RFC 822*, IETF, August 1982.

[4] Fabio Bellifemine, Agostino Poggi, and Giovanni Rimassa, "JADE: a FIPA2000 compliant agent development environment," in *Proceedings of the Fifth International Conference on Autonomous Agents*, Jörg P. Müller, Elisabeth Andre, Sandip Sen, and Claude Frasson, Eds. 2001, pp. 216–217, ACM Press.

[5] S. Posland, P. Buckle, and R. Hadingham, "The FIPA OS agent platform: Open source for open standards," in *Proceedings of the 5th International Conference and Exhibition on the Practical Application of Intelligent Agents and Multi-Agents*, Jeffrey Bradshaw and Geoff Arnold, Eds., UK, 2000, pp. 355–368.

[6] J. Callas, L. Donnerhacke, H. Finney, and R. Thayer, *OpenPGP Message Format, IETF RFC 2440*, IETF, November 1998.

[7] Philip R. Zimmermann, *The Official PGP User's Guide*, MIT Press, Boston, 1995.

[8] R. Housley, *Cryptographic Message Syntax, IETF RFC 3369*, IETF, August 2002.

[9] B. Ramsdell, *S/MIME Version 3 Message Specification, IETF RFC 2633*, IETF, June 1999.

[10] RSA Laboratories, *PKCS #7: Cryptographic Message Syntax Standard*, 1993, version 1.5.